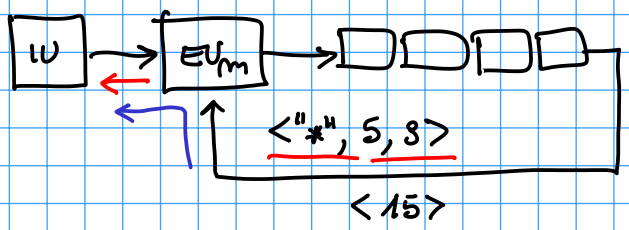
* int

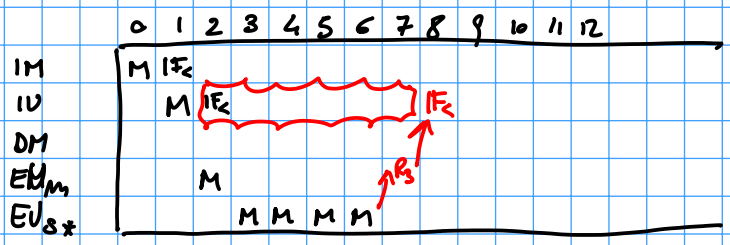


MUL R_1, R_2, R_3

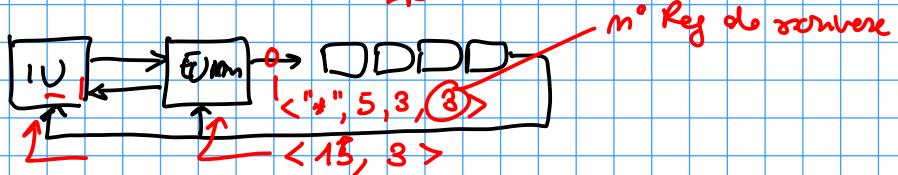
↑ ↑

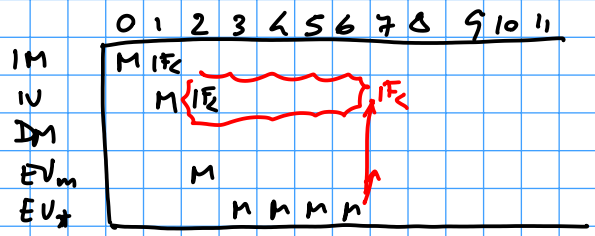
5 3

IF< $R_3, R_N,$ loop

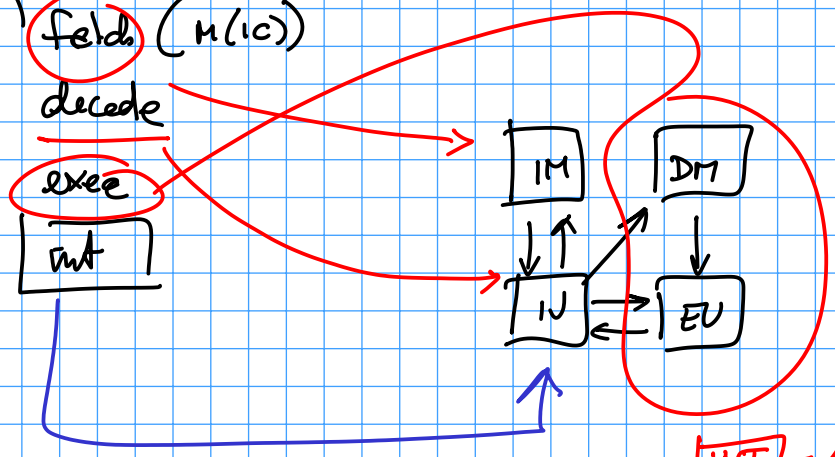


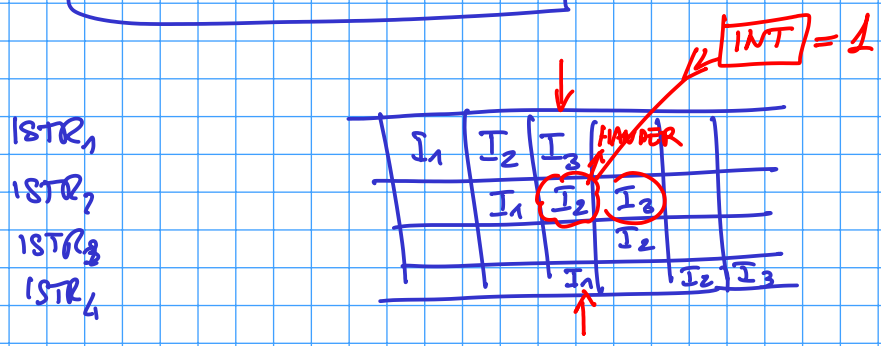
15





```
while(true) {
    fetch (M(IC))
    decode
    exec
    mt
}
```



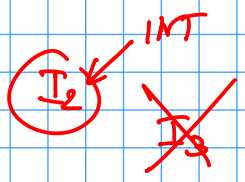


interruzioni

IMPRECISE

→ INT ⇒ call (salto)

PRECISE



ARCHITETTURA SUPERSCALARE (n-vie)

$\forall t$: lavoriamo su n istruzioni diverse

n=2

IM farà il fetch di 2 istr

IU decodificherà 2 istr

DM_1 sarà in grado di eseguire 2 op LD/ST

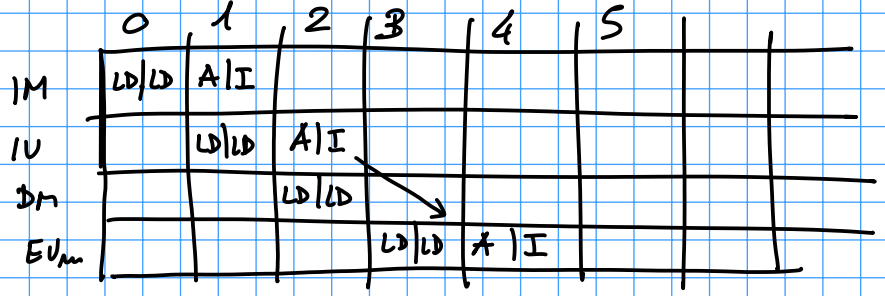
EU_{DM} sarà in grado di eseguire 2 operaz AL calcolo schedulare 2 operaz. AL luogo della unità slave

```
LD    Rb, Ri, R1
LD    Rb', Ri', R2
ADD   R1, R2, R3
INC   Ri
```

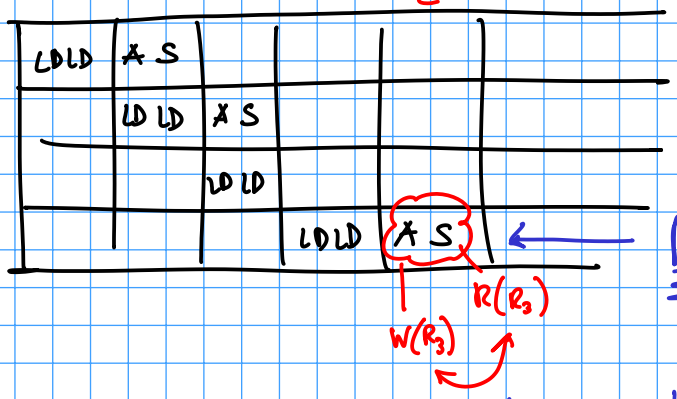
PIPELINE

	0	1	2	3	4	5	6	7	8
IM	L	L_2	A	I					
IU		L	L_2	A	I				
DM			L	L_2					
EU_{DM}				L	L_2	A	I		

LD_1	LD_2
ADD	INC




```
LD    Rb, Ri, R1
LD    Rb', Ri', R2
ADD   R1, R2, R3
SUB   R1, R3, R4
```



1) non devo avere dip IU-EU fra le due istruzioni "della slot"

2) non devo avere due istruzioni di salto

3) non devo avere dip EU-EU

```
LD   Rb, Ri, R1
LD   Rb', Ri, R2
ADD  R1, R2, R3
SUB  R4, R3, R4
```

t	LD_1	LD_2
t	ADD	NOP
t	SUB	NOP

pipeline da 6 istr

⇓

4 istr + 2 NOP

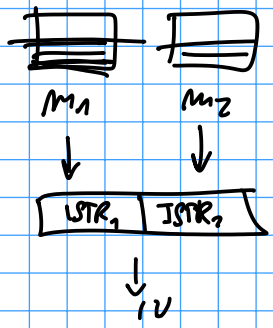
$T_{id} = 2t$

$T_c = 3t$

$\varepsilon = \frac{2}{3}$

IM fetch di 2 istr alla volta

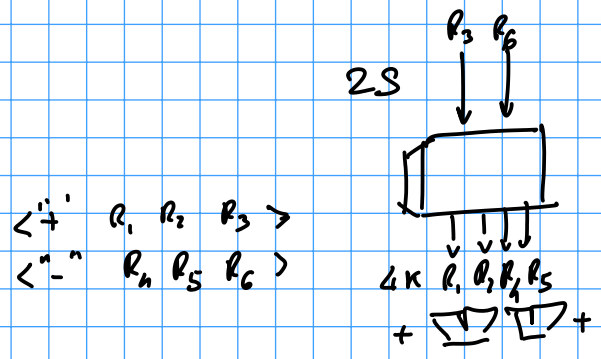
mem modulare interlacciata



IU decodifica 2 istruz alla volta

DM

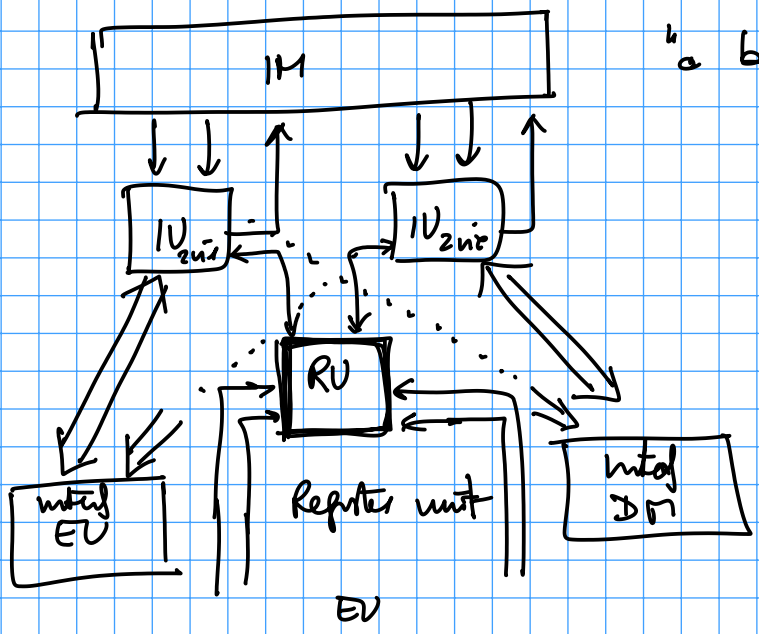
EU



$< \text{'+'} \; R_1 \; R_2 \; R_3 >$

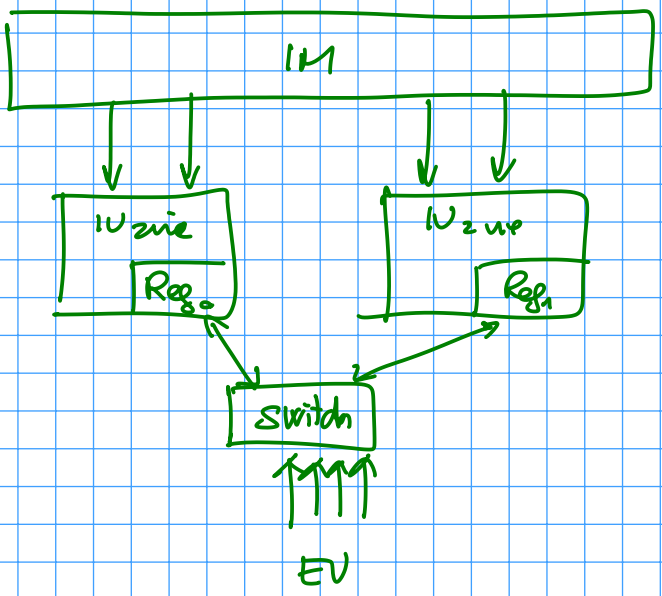
$< \text{'-'} \; R_4 \; R_5 \; R_6 >$

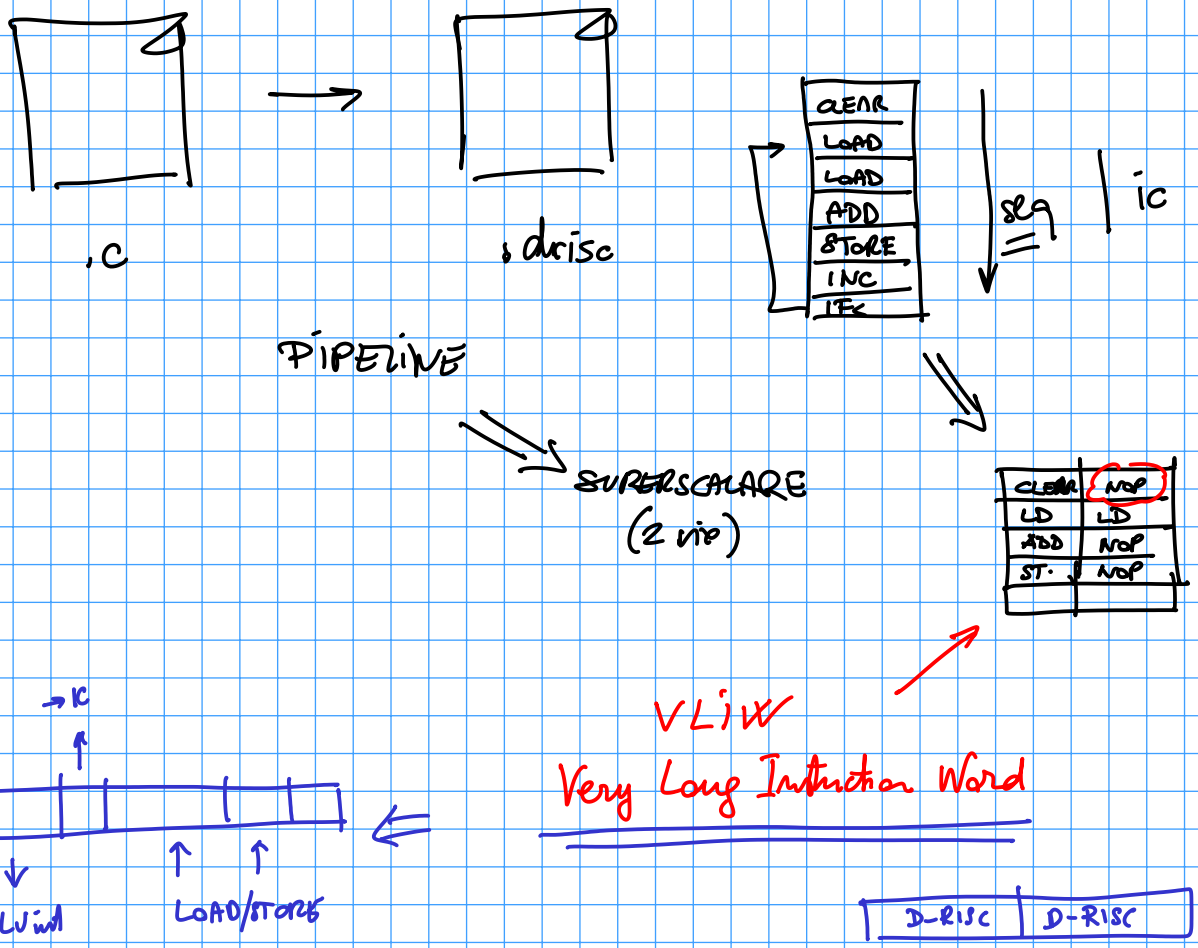
IU 4 vie (bloccki costituiti IU a 2 vie "a banda larga"



Con IU parzialmente replicata

Con IU completamente replicata




.c
.d-risc
CLEAR
LOAD
LOAD
ADD
STORE
INC
IFK
seq
IC
PIPELINE
SUPERSCALARE
(2 via)
CLEAR
NOP
LD
LD
ADD
NOP
ST.
NOP
stream di coppie di istruzioni d-RISC
VLIW
Very Long Instruction Word
IC
ALU int
LOAD/STORE
D-RISC
D-RISC

PROC D-RISC

Seq

(interprete fw)

1 iterazione del ciclo

while (true) {F, D, E, INT}

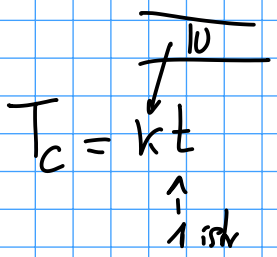
alla volta

$$T_c \quad \sum \left(T_{fo} + T_{do} + T_{exec_i} \right)$$

PROC D-RISC

Pipeline

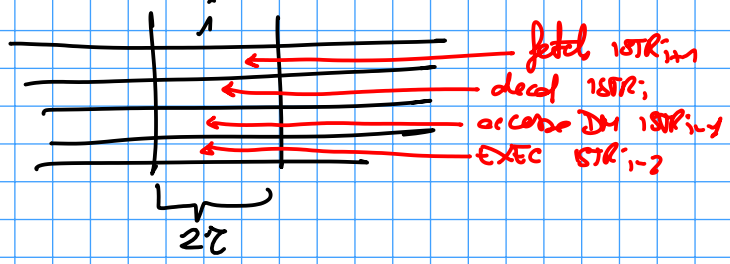
(EU | EU_m + EU_{slave})



fasi diverse del ciclo F-D-E-INT

relative a istruzioni diverse

avvengono contemporaneamente

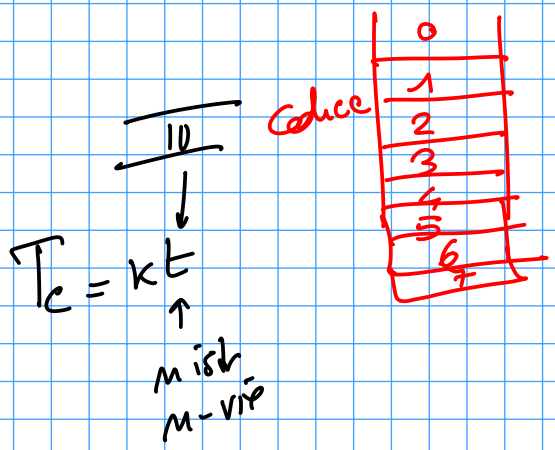


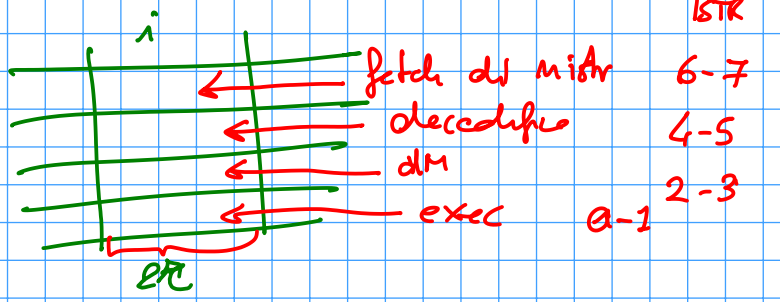
PROC D-RISC

pipeline superscalare

come nel pipeline i fasi diverse di (M coppie) istruzioni diverse

eseguite contemporaneamente





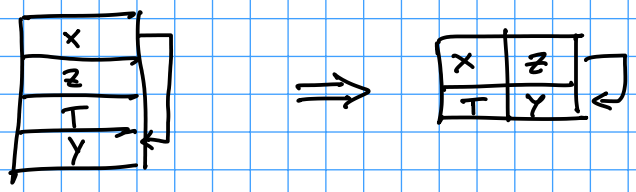
PIPELINE D-RISC

- problema dei salti
- problema delle dip logiche IU-EU
- problema delle dip logiche EU-EU

↓

PIPELINE D-RISC superscalare

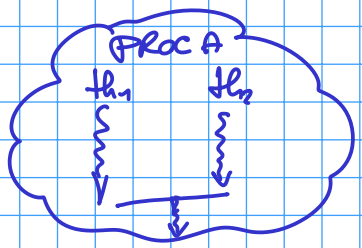
probl delle dip si accentua



MULTITHREADING

THREAD flusso di controllo (seq di istruzioni) di un certo processo

THREAD diversi possono eseguire concorrentemente



A[N] B[N] media delle diff $|a[i] - b[i]|$

```
sum = 0;
for (i=0; i<N; i++)
    sum += abs(a[i] - b[i]);
media = sum / N;
```

```
sum0 = 0                    sum1 = 0
for(i=0; i<N/2; i++)        for(i=N/2; i<N
    sum0 += ...                 sum1 += ...
sum = sum0 + sum1;
media = sum / N;
```

```
    CLEAR  Ri
    CLEAR  Rsum
loop: LOAD1  RbaseA, Ri, R1
    LOAD   RbaseB, Ri, R2
    SUB    R1, R2, R3
    IF>=0  R3, cont
    SUB    R0, R3, R3
cont: ADD  Rsum, R3, Rsum
    INC    Ri
    IF<    Ri, RN, loop
next:
```

multithread interleaved

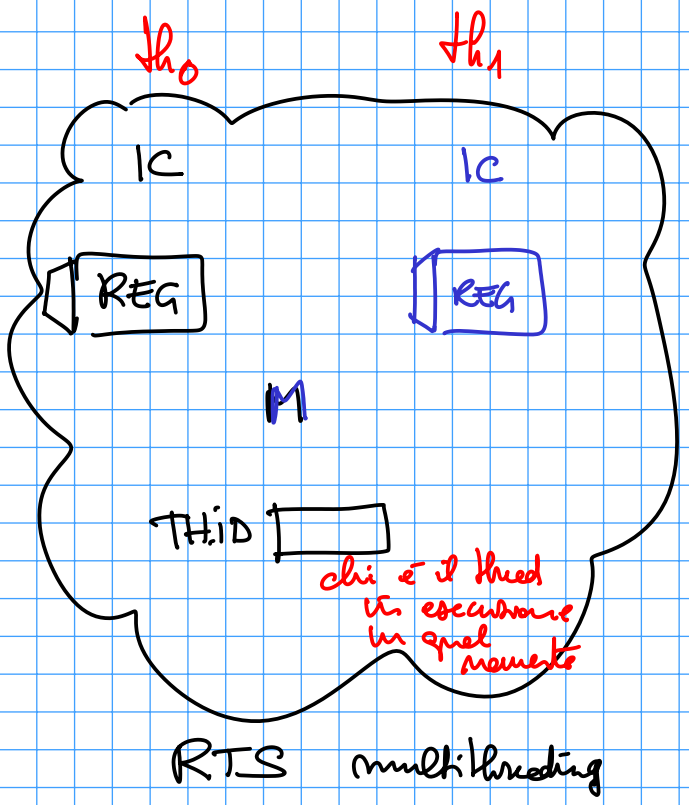
```
LOAD1
LOAD1
LOAD2
LOAD2
SUB
SUB
IF>=0
IF>=0
```

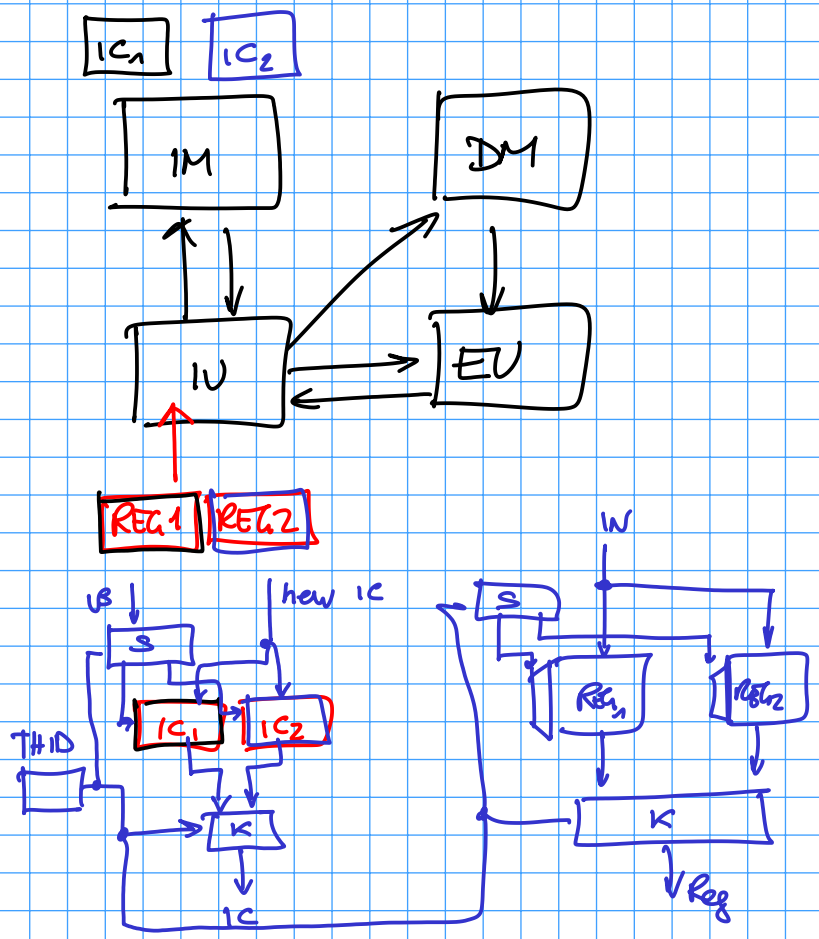
multithread blocking

```
LOAD1
LOAD2
SUB
IF>=0
LOAD1
LOAD2
SUB
IF>=
IF>=
SUB
ADD
```

dip → blocco/bolle

dip → blocco





interleaved

$\forall t \quad THID \leftarrow (THID + 1) \% 2$

blocking $\forall$ volta che IU tenta di leggere un reg col CNT $\neq \emptyset$

$THID \leftarrow (THID + 1) \% 2$

2 thread multithreading a 2 vie

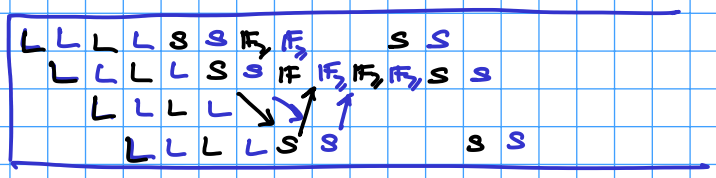
$\Downarrow$

generalizzato a n vie $(n = 2^k)$

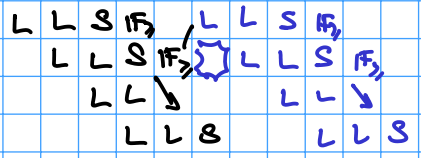
```
loop: LOAD1  R_baseA, Ri, R1   1
      LOAD   R_baseB, Ri, R2   2
      SUB    R1, R2, R3        3
      IF>0   R3, cont          4
      SUB    R0, R3, R3        5
cont: ADD    R_sum, R3, R_sum  6
      INC    Ri                7
      IF<    Ri, RN, loop      8
```

Th1 Th2

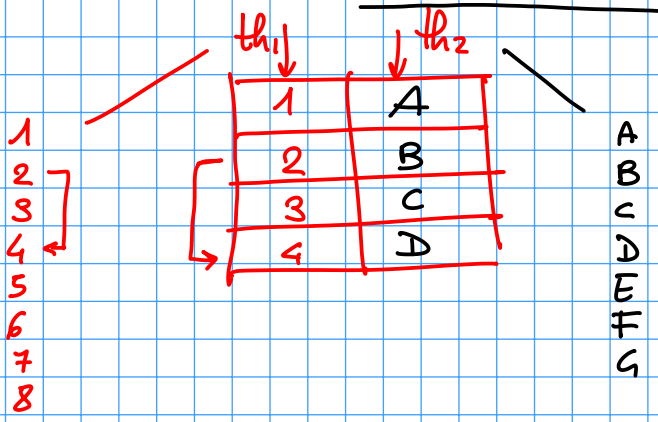
interleaving a 2 vie



blocking



D-RISC PIPELINE SUPERSCALARE a 2 vie



x multithreading a 2 vie

⇓⇓

hyperthreading